Review: Solitaire: Man Versus Machine

Xiang Yan

Persi Diaconis

Paat Rusmevichientong StanfordUniversity

xyan,persi.diaconis,bvr@stanford.edu

Benjamin Van Roy Cornell University

paatrus@orie.cornell.edu

1 Introduction

The main objective of this paper is to design a strategy to improve the success rate in klondike solitaire. So, a rollout method is implemented on the given heuristic for the same. This strategy using iterated roll-out wins about twice as many games on average as an expert player does.

2 Strategy

Klondike is played with a standard deck of cards. There are thirteen stacks, a pile, a talon and four suit stacks as shown in the figure. The goal of the game is to move all cards into suit stack starting from aces, then two's and so on. However, in this version the identity of each card at each position is revealed to the player at the beginning of the game. This is known as thoughtful solitaire, however usual rules of klondike solitaire still apply. The objective is to end up all cards on suit stack.

General procedure during machine play:

a. Identify set of legal moves.
b. Select and execute a legal move.
c. If all cards are on suit stacks, declare victory and terminate
d. If new card configuration repeats a previous one, declare loss and terminate.
e. Repeat procedure.

Heuristic Strategy:

This follows the general scoring rules of the klondike solitaire however in our strategy, a score is assigned to each move and zero is assigned to a move not covered in the scoring rules. So, we basically look for the move which can give a maximum score as an output. However, there might be some situation where multiple moves maximizes the score, So to break that following priority rules are applied:

Case1: Card is moved from build stack to another build stack

a. Let k be the number of originally face-down card. So, if the move turns and face-down card face-up it gets a priority of k+1
b. If the moves empty the stack priority of 1 is assigned.

Case2: Card is moved from talon to build stack

a. If card moved is king the priority is 1
b. If card moved is king and matching queen is in the pile or talon or suit stack or face-up in build stack the priority of 1 is assigned.
c. If card moved is king and its matching queen is face-down in build stack the priority of -1 is assigned.

Case3: For any card move not covered above we assign a priority of Zero.

Moreover, a little bit of modification is also introduced in scoring system. if card being moved is a king and its matching queen is face-down in build stack we assign a score of 0.

Rollout:

Rollout method is used to amplify the performance of any strategy. If there is an strategy h then this procedure generates another strategy h', which is called as rollout strategy.

So, if a card configuration is given let say x. A strategy h will make a move h(x). Then a rollout strategy makes a move h'(x) determined by following:

1. For each legal move a, simulate the remainder of the game, taking move a and then employing strategy h thereafter
2. If any of these simulations leads to victory, chose one of them randomly and let h'(x) be the corresponding move a.
3. If none of the simulations lead to victory, let h'(x)=h(x).

This procedure can be further iterated to generates another improved strategy h''. So, after a finite number of such iteration, we can get an optimal strategy.

3 Evaluation

3.1 PROS

1. This version of solitaire is easy to understand and improves the success rate.

2. Success rate becomes more than double using the sufficient number of rollouts.

3. The introduction of priority moves improves the gameplay, for example, if we are in a situation where we can move either one of the blocks on two separate build stacks, we prefer to move the block which have more face-down cards as it will have a high priority.

4. So, this will finally strive to balance the number of originally face-down cards. This clearly improves the success rate as it involves much more thinking.

3.2 Cons

1. There is no method address about the time complexity of the game. For example, let say we are playing the game with a certain time limit. So, our method will find a best strategy to maximize the score but it will not find the best strategy that can maximize the score within the time limit of the game. So always getting the highest score is not always the case.

2. The heuristic strategy has practically disabled the card moves from a suit stack to a build stack. But there might be some cases where this move would lead to a new remainder which might give a maximum score at the end. This will lead to a loss in game and failure in the method.

3. Our rollout method requires a sufficient number of iterations but this may not be feasible in every case. Computational requirements grow after each iteration. So prior application of the rollout method has a single iteration. But the best output as seen in results was obtained after 3-4 iterations.

4 Results

We randomly generated a large number of games and player them with our algorithm. For the original heuristic a confidence bound of [-1.4%,1.4%] was achieved. For 4 and 5 rollout we obtained a weaker confidence bound due to time constraints, However the success rate became twice as human expert play.

The Results obtained are shown in the image attached. As we can see that the success rate increases as number of rollouts increased and is almost twice as more as human expert moreover the confidence bound also increased.

Player	Success Rate	Games Played	Average Time Per Game	99% Confidence Bounds
Human expert	36.6%	2,000	20 minutes	±2.78%
heuristic	13.05%	10,000	.021 seconds	±.882%
1 rollout	31.20%	10,000	.67 seconds	±1.20%
2 rollouts	47.60%	10,000	7.13 seconds	±1.30%
3 rollouts	56.83%	10,000	1 minute 36 seconds	±1.30%
4 rollouts	60.51%	1,000	18 minutes 7 seconds	±4.00%
5 rollouts	70.20%	200	1 hour 45 minutes	±8.34%